RECEIVED
CENTRAL FAX CENTER
FEB 2 3 2007

CURRENT LISTING OF CLAIMS

This listing of claims replaces all prior versions, and listings, of claims in the application:

1. (Original) A database system comprising:

a storage to store a view containing results of a cube-based operation on at least one base table, the view containing a first result set for a group-by on a first grouping set, and a second result set for a group-by on a second grouping set; and

a controller, in response to a change to the at least one base table, to:

update the first result set by computing a change to the first result set based on a change in the at least one base table; and

update the second result set by computing a change to the second result set based on the change to the first result set.

2. (Original) The database system of claim 1, wherein the first grouping set has a first number of grouping attributes, and the second grouping set has a second number of grouping attributes, the first number being greater than the second number.

3. (Original) The database system of claim 2, wherein the view further contains a third result set for a group-by on a third grouping set having a third number of grouping attributes, the third number less than the second number,

the controller to further update the third result set by computing a change to the third result set based on the change to the second result set.

4. (Original) The database system of claim 1, wherein the view contains results of a group-by cube operation.

5. (Original) The database system of claim 1, wherein the view contains results of a group-by partial cube operation.

6. (Original) The database system of claim 1, further comprising plural access modules and plural storage modules, the access modules to enable parallel access of data in the plural storage modules.

7. (Original) The database system of claim 6, wherein the controller is adapted to distribute rows in the first result set across the access modules based on a hash of columns of the second grouping set and at least another column that is assigned a predefined value.

8. (Original) The database system of claim 7, wherein the view contains results of a cube operation specified by a cube function on plural columns, the at least another column being one of the plural columns of the cube function that is not in the second grouping set.

9. (Original) The database system of claim 7, wherein the view is distributed across the access modules such that plural portions of the view reside in respective storage modules, and wherein the rows in the first result set are distributed across the access modules according to the hash to enable:

each access module to locally perform a merge and aggregate operation on the rows of the first result set to produce rows of the second result set; and

each access module to locally merge the rows of the second result set into a respective portion of the view without having to first redistribute the rows of the second result set.

10. (Original) The database system of claim 1, wherein the controller is adapted to further:

receive a query specifying a group-by operation; and

determine whether an answer for the query specifying the group-by operation can be satisfied from the view.

11. (Original) The database system of claim 10, wherein the query specifies a group-by operation on grouping sets S, and the view contains result sets for grouping sets C,

the controller to determine whether S is a subset of C to determine whether the answer for the query can be satisfied from the view.

12. (Original) The database system of claim 11, wherein the controller is adapted to modify a WHERE clause of the query in response to determining that S is a subset of C.

13. (Original) A method for use in a database system, comprising:

storing a view containing results of a cube-based operation on at least one base table, the view containing result sets for group-bys on respective grouping sets;

updating a first result set by computing a change to the first result set based on a change in the at least one base table; and

updating a second result set by computing a change to the second result set based on the change to the first result set.

14. (Original) The method of claim 13, wherein updating the first result set comprises updating the first result set for the group-by on a first grouping set that has a greater number of columns than a second grouping set corresponding to the second result set.

15. (Original) The method of claim 13, further comprising updating a third result set by computing a change to the third result set based on the change to the second result set.

16. (Original) The method of claim 15, further comprising updating a fourth result set by computing a change to the fourth result set based on the change to the third result set.

17. (Original) The method of claim 13, wherein the database system has plural storage modules to store respective portions of the view, and plural access modules to manage access of respective storage modules,

wherein updating the first result set and second result set are performed in parallel by the plural access modules.

18. (Original) The method of claim 17, further comprising distributing rows of the first and second result sets across the plural access modules.

19. (Previously Presented) The method of claim 18, wherein the first result set corresponds to a group-by on a first grouping set having N columns, and the second result set corresponds to a group-by on a second grouping set having N-1 columns, and wherein distributing the first result set to compute the second result set comprises distributing the first result set based on a hash of the N columns, with the column in the first grouping set not present in the second grouping set being assigned a predefined value.

20. (Previously Presented) The method of claim 19, further comprising:

updating a third result set by computing a change to the third result set based on the change to the second result set, wherein the third result set corresponds to a group-by on a third grouping set having N-2 columns,

wherein distributing the second result set across the access modules to compute the third result set is based on a hash of the N columns, with the columns in the first grouping set not appearing in the third grouping set each being assigned to the predefined value.

21. (Original) The method of claim 20, wherein storing the view comprises storing a view for a cube operation based on a cube function of the N columns.

22. (Original) An article comprising at least one storage medium containing instructions that when executed cause a database system to:

store a view containing results of a cube-based operation on at least one base table, the view containing result sets for group-bys on respective grouping sets;

update a first result set by computing a change to the first result set based on a change in the at least one base table; and

update a second result set by computing a change to the second result set based on the change to the first result set.

23. (Original) The article of claim 22, wherein updating the first result set comprises updating the first result set for the group-by on a first grouping set that has a greater number of columns than a second grouping set corresponding to the second result set.

24. (Original) The article of claim 22, wherein the instructions when executed cause the database system to further update a third result set by computing a change to the third result set based on the change to the second result set.

25. (Original) The article of claim 22, wherein the database system has plural storage modules to store respective portions of the view, and plural access modules to manage access of respective storage modules,

wherein updating the first result set and second result set are performed in parallel by the plural access modules.

26. (Original) The article of claim 25, wherein the instructions when executed cause the database system to further distribute rows of the first and second result sets across the plural access modules.

27. (Previously Presented) The article of claim 26, wherein the first result set corresponds to a group-by on a first grouping set having N columns, and the second result set corresponds to a group-by on a second grouping set having N-1 columns, and wherein distributing the first result set to compute the second result set comprises distributing the first result set based on a hash of the N columns, with the column in the first grouping set not present in the second grouping set being assigned a predefined value.

28. (Original) The article of claim 27, wherein storing the view comprises storing a view for a cube operation based on a cube function of the N columns.